

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In The Application Of:
Kei NAKAYAMA and Kotaro TERADA

Serial No.: Not Yet Assigned

Filing Date: Concurrently Herewith

For: DIGITAL MIXER AND CONTROL
METHOD FOR DIGITAL MIXER

Examiner: Not yet assigned

Group Art Unit: Not yet assigned

TRANSMITTAL OF PRIORITY DOCUMENT

Mail Stop Patent Application
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

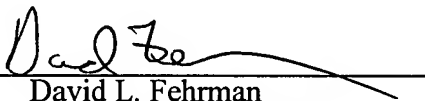
Dear Sir:

Enclosed herewith is a certified copy of Japanese Patent Application No. 2002-210310 filed July 18, 2002, from which priority is claimed under 35 U.S.C. 119 and Rule 55.

Acknowledgement of the priority document is respectfully requested to ensure that the subject information appears on the printed patent.

Dated: July 18, 2003

Respectfully submitted,

By: 
David L. Fehrman
Registration No. 28,600

Morrison & Foerster LLP
555 West Fifth Street
Suite 3500
Los Angeles, California 90013-1024
Telephone: (213) 892-5601
Facsimile: (213) 892-5454

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 7月18日

出 願 番 号

Application Number:

特願2002-210310

[ST.10/C]:

[JP2002-210310]

出 願 人

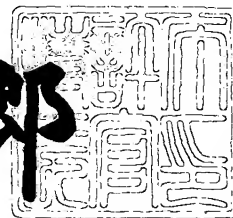
Applicant(s):

ヤマハ株式会社

2003年 5月23日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田 信一郎



出証番号 出証特2003-3037970

【書類名】 特許願

【整理番号】 DY677

【あて先】 特許庁長官殿

【国際特許分類】 G10K 15/00

【発明者】

 【住所又は居所】 静岡県浜松市中沢町 1 0 番 1 号 ヤマハ株式会社内

 【氏名】 中山 圭

【発明者】

 【住所又は居所】 静岡県浜松市中沢町 1 0 番 1 号 ヤマハ株式会社内

 【氏名】 寺田 光太郎

【特許出願人】

 【識別番号】 000004075

 【氏名又は名称】 ヤマハ株式会社

【代理人】

 【識別番号】 100096954

 【弁理士】

 【氏名又は名称】 矢島 保夫

【手数料の表示】

 【予納台帳番号】 022781

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 デジタル・ミキサ

【特許請求の範囲】

【請求項 1】

複数の機能ブロックが共働して動作することによりミキサとして動作するデジタル・ミキサであって、

電源立ち上げ時に、前記複数の機能ブロックのバージョンを取得する手段と、
前記取得した各機能ブロックのバージョンの整合性をチェックする手段と、
不整合がなければ動作を継続し、不整合があればユーザに警告する手段と
を備えたことを特徴とするデジタル・ミキサ。

【請求項 2】

複数の機能ブロック毎に CPU と該 CPU が実行する制御プログラムを記憶した記憶手段を備え、該複数の機能ブロックが共働して動作することによりミキサとして動作するデジタル・ミキサであって、

電源立ち上げ時に、前記複数の機能ブロックの制御プログラムのバージョンを取得する手段と、

前記取得した各制御プログラムのバージョンの整合性をチェックする手段と、
不整合がなければ動作を継続し、不整合があればユーザに警告する手段と
を備えたことを特徴とするデジタル・ミキサ。

【請求項 3】

請求項 1 または 2 に記載のデジタル・ミキサにおいて、

前記バージョンの不整合があった場合、その不整合が全体の動作にとって致命的であるか否か判定する手段と、

致命的であるときは動作を停止し、致命的でないときは、バージョンの不整合を無視して動作を継続する旨の通知を行なうとともに、不整合と判定された機能ブロックを切り離して動作を継続する手段とを

さらに備えたことを特徴とするデジタル・ミキサ。

【請求項 4】

請求項 1 または 2 に記載のデジタル・ミキサにおいて、

新たな機器が接続された場合に、該機器の各機能ブロックのバージョンを取得してその整合性をチェックする手段と、

バージョンの不整合があった場合、その不整合が全体の動作にとって致命的であるか否か判定する手段と、

致命的であるときは動作を停止し、致命的でないときは、前記バージョンの不整合を無視して動作を継続する旨の通知を行なうとともに、不整合と判定された機能ブロックを切り離して動作を継続する手段とを

さらに備えたことを特徴とするデジタル・ミキサ。

【請求項 5】

請求項 1 または 2 に記載のデジタル・ミキサにおいて、

前記各機能ブロックの異常を検出する手段と、

異常が検出された場合、その異常が全体の動作にとって致命的であるか否か判定する手段と、

致命的であるときは動作を停止し、致命的でないときは、前記異常を無視して動作を継続する旨の通知を行なうとともに、異常と判定された機能ブロックを切り離して動作を継続する手段とを

さらに備えたことを特徴とするデジタル・ミキサ。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

この発明は、音響設備の集中的な制御を行なうデジタル・ミキサにおけるバージョン管理技術に関する。

【 0 0 0 2 】

【従来の技術】

従来より、コンサートや演劇などを行なう会場において音響設備を制御するためのデジタル・ミキサが知られている。そのような音響設備では、多数のマイクロフォンおよび多数のスピーカが使用され、効果音なども多種多様に使用される。デジタル・ミキサは、多数の入力をどのようにミキシングして、どのように効果を付与し、どの出力系統に出力するか、などを集中的に制御する。

【0003】

このようなデジタル・ミキサとして、コンソールとエンジンとを分離し、コンソールをオペレータ側に配置し、エンジンを演奏者側に配置し、それらの間をケーブルで接続するものが知られている（例えば、特開2000-293180）。このようなミキサでは処理量が多いため、複数の機能ブロックに分けて、それぞれを1つあるいは複数のCPUで制御するようになっている。複数の機能ブロックが共働してミキサとしての全体の動作を実現する。各機能ブロックのCPUで実行するそれぞれの制御プログラム（いわゆるファームウェア）は、各機能ブロックごとに設けられた所定の記憶領域上に記憶されている。それらの各機能ブロック毎の複数のプログラムは、それぞれがバージョンを管理されており、それぞれバージョンアップすることができるようになっている。

【0004】

また、特願2000-396105では、コンソールとエンジン分離型のミキサにおいて、コンソールあるいはエンジンのいずれかにPC（パーソナルコンピュータ）を接続し、そのPCでバージョンアッププログラムを実行し、コンソールがマスタとなって各機器のバージョンをチェックするシステムが記載されている。

【0005】

【発明が解決しようとする課題】

上述のコンソールとエンジン分離型のミキサにおけるバージョンアップ方式では、一度のバージョンアップ操作で、そのとき接続されている機器の全機能ブロックの制御プログラムを同時にバージョンアップできる。

【0006】

しかし、この方式を用いたとしても、ミキサの一部の機能ブロックのバージョンが他のブロックのバージョンと整合しない場合がある。例えば、バージョンアップ操作時に接続されていなかった機器内のブロックについてはバージョンアップされない。あるいはその機器を後から接続した場合、当該機器内のブロックだけバージョンが合わないことがある。また、一部のブロックの故障が検出され、そのブロックのボードを新しいものに交換した場合も、そのボードの制御プログ

ラムのバージョンが他の部分と整合しない場合がある。制御プログラムのバージョンに不整合がある状態で動作させると、誤動作をしたり、場合によっては機器を壊してしまう可能性がある。

【 0 0 0 7 】

さらに、制御系統の機能ブロックについては、その中の一部が故障したとしても、全システムが動作停止してしまうという問題があった。すなわち、制御系統については、全ブロックが正常に動作していないと全体として正常動作できなかった。

【 0 0 0 8 】

この発明は、複数の機能ブロックを備え、各機能ブロック毎にCPUとそのCPUが実行する制御プログラムを備えているようなデジタル・ミキサにおいて、それら各機能ブロック毎のCPUで動作する制御プログラムのバージョンを、適正にかつ容易に管理できるようにすることを目的とする。

【 0 0 0 9 】

【課題を解決するための手段】

上記目的を達成するため、この発明では、電源立ち上げ時に、システムの全ブロックのバージョンの整合性をチェックし、不整合がなければ動作を継続し、不整合があった場合はユーザに警告する。その警告において、不整合がシステムの動作にとって致命的である場合は動作を停止し、致命的でない場合はユーザに「バージョンの不整合を無視して動作を継続する」という内容の通知をした後、不整合と判定されたブロックを切り離して動作を継続する。

【 0 0 1 0 】

新たな機器が接続された場合にも、その接続された機器の各ブロックについてのバージョンチェックを行なう。動作上不可欠な機器が接続された場合、新たに接続された機器のバージョンの整合性をチェックし、不整合がなければ動作を継続し、不整合があった場合は動作を停止するとともにユーザに警告する。動作上不可欠でない機器が接続された場合、新たに接続された機器のバージョンの整合性をチェックし、不整合がなければ動作を継続し、不整合があった場合は該機器との連結動作を中止し動作を継続するとともにユーザに警告する。

【 0 0 1 1 】

また、この発明では、各ブロックについて故障を検出し、その故障がシステム動作にとって致命的である場合は動作を停止し、致命的でない場合はユーザに「故障を無視して動作を継続する」という内容の通知をした後、故障したブロックを切り離して動作を継続する。

【 0 0 1 2 】

【発明の実施の形態】

以下、図面を用いてこの発明の実施の形態を説明する。

【 0 0 1 3 】

図 1 は、この発明の実施の形態に係るデジタル・ミキサの構成例を示す。101 はコンソール、102 はエンジン、103 は各ユニットを示す。コンソール101 は、ユーザが操作する多数の操作子やディスプレイを備えた部分であり、ユーザによる操作指示を受け付け、エンジン102 に指示を出す。エンジン102 は、コンソール101 からの指示に応じて、入力ユニットから入力したマイク信号やライン信号などを適宜ミキシングし、効果の付与などを行なって、出力ユニットに出力する。ユニット103 としては、アナログ入力（A I N）ユニット、アナログ出力（A O U T）ユニット、およびデジタル入出力（D I / O）ユニットが接続される。A I Nユニットは、1 ユニットに複数枚のアナログ／デジタル（A / D）変換ボードをマウントできるボックスであり、これによりマイク信号やライン信号を複数チャンネル（c h）分入力することができる。A O U Tユニットは、1 ユニットに複数枚のデジタル／アナログ（D / A）変換ボードをマウントできるボックスであり、複数 c h 分の出力を行なうことができる。D I / Oユニットは、1 ユニットに複数枚のデジタル入出力（I / O）ボードをマウントできるボックスであり、複数 c h 分のデジタル入力およびデジタル出力を行なうことができる。

【 0 0 1 4 】

1 台のエンジン102 には、入力側に10 台のユニット（A I NユニットまたはD I / Oユニット）を接続でき、出力側に6 台のユニット（A O U TユニットまたはD I / Oユニット）を接続できる。1 台のコンソール101、1 台のエン

ジン 1 0 2、および当該エンジン 1 0 2 に接続された複数のユニット 1 0 3 の組み合わせが、ミキサとしての 1 システムの最小構成となる。これにより、任意の入力 c h の入力信号を任意のミキシング c h に流し込んでミキシングし、任意に効果付与して、任意の出力 c h に出力する処理を、集中的に制御できる。

【 0 0 1 5 】

図 1 の①は、コンソール 1 0 1 にもう 1 台のコンソール 1 1 1 を接続し、デュアルコンソールとした例である。これにより、操作子の数を増やし多数の c h についての直接的な制御環境を構築したり、1 台のコンソールが故障した場合に他の 1 台のコンソールで操作を続行できるようなフェイルセーフシステムを構築できる。デュアルコンソールとした場合、初めからあるシステムのコンソール 1 0 1 はマスタコンソール、後から接続したコンソール 1 1 1 はスレーブコンソールと呼ぶ。コンソール 1 0 1 内の MM の記載は、マスタシステムのマスタコンソールであることを示す。コンソール 1 1 1 内の M S の記載は、マスタシステムのスレーブコンソールであることを示す。

【 0 0 1 6 】

図 1 の②は、エンジンのミラーリングの例である。ミラーリングでは、マスタシステムのコンソール 1 0 1 とユニット 1 0 3 に、もう 1 台のエンジン 1 2 1 を接続する。ミラーリングでは、コンソール 1 0 1 からの制御により、エンジン 1 0 2 と 1 2 1 は同じ動作を行なう。通常は選択されているエンジン（例えば 1 0 2）が使われるが、当該エンジンに故障があったとき、もう一方のエンジンに切り換えて処理を継続できる。

【 0 0 1 7 】

図 1 の③は、システムのカスケード接続の例である。コンソール 1 3 1、エンジン 1 3 2、およびユニット 1 3 3 を別システムとして用意し、マスタシステムのエンジン 1 0 2 とエンジン 1 3 2 とをカスケード接続する。コンソール 1 3 1、エンジン 1 3 2、およびユニット 1 3 3 でスレーブシステムを構成する。これにより、2 台のエンジン 1 0 2、1 3 2 のそれぞれのミキシングバスが連結され、全体として入力 c h 数が倍のミキサシステムが構築できる。なお、コンソール 1 3 1 内の S M の記載は、スレーブシステムのマスタコンソールであることを示

す。エンジン 1 3 2 内の S の記載は、スレーブシステムのエンジンであることを示す。

【 0 0 1 8 】

図 2 は、1 台のコンソールの外部パネルの例を示す。外部パネルは、2 0 1 ～ 2 1 0 の各セクションに分かれている。そのうち、セクション 2 0 1 ～ 2 0 5, 2 0 7 ～ 2 0 9 は、スイッチ、フェーダ、およびロータリーエンコーダなどの複数の操作子を備えた部分である。

【 0 0 1 9 】

2 0 1, 2 0 2, 2 0 3, 2 0 4 は、それぞれが 1 つの入力 c h セクション (I N A ～ D) である。入力 c h セクションは、複数入力 c h についての各種制御を行なう部分である。2 0 5 はセレクトッド入力 c h セクション (I N S E L) であり、入力 c h セクション 2 0 1 ～ 2 0 4 で選択した 1 つの入力 c h について各種の制御を行なう部分である。表示器 (L C D) 2 0 6 は、各種の情報を表示する液晶表示装置とマウス操作子のインターフェースを備えた部分である。出力 c h セクション (O U T) 2 0 7 は、複数の出力 c h についての各種制御を行なう部分である。セレクトッド出力 c h セクション (O U T S E L) 2 0 8 は、出力 c h セクション 2 0 7 で選択した 1 つの出力 c h について各種の制御を行なう部分である。シーンセクション (S C N) 2 0 9 は、シーンに関する各種の制御を行なう部分である。シーンとは、ある場面での設定状態を示す。例えば、各場合毎の設定をシーンとして記憶させておき、それを呼び出すことにより設定した状態を簡単に再現できる。メーター (M T R) 2 1 0 は、各 c h のレベルを示すメーターなどを備えた部分である。

【 0 0 2 0 】

図 3 に、1 台のコンソールの内部構成例を示す。コンソール 3 0 0 は、C P U 3 0 1、フラッシュメモリ (C M A I N) 3 0 2、ランダムアクセスメモリ (R A M) 3 0 3、通信制御部 (O C O M) 3 0 4、コントロール部 (C T L) 3 0 5、表示器 (L C D) 3 0 6、パネル制御部 A (P N L A) 3 0 7、パネル制御部 B (P N L B) 3 0 8、波形通信制御部 (W C O M) 3 0 9、通信制御部 (C O M) 3 1 0、セレクトッド入力 c h セクション (I N S E L) 3 2

1、セレクトッド出力c hセクション (OUT SEL) 3 2 2、シーンセクション (SCN) 3 2 3、出力c hセクション (OUT) 3 2 4、入力c hセクション (IN A~D) 3 2 5~3 2 8、メーター (MTR) 3 2 9、および接続バス3 1 1を備えている。なお、接続バス3 1 1は、1つのバスラインで図示したが、実際には各部が相互に情報をやり取りするための複数の方式を適用している。また、図2の各セクションと図3の各ブロックとで、対応する部分は同じ記号で図示している。例えば、図3のIN A 3 2 5は図2のセクション2 0 1に対応している。

【0 0 2 1】

図3において、太枠で図示したブロックは、それぞれが、その内部にCPUおよびフラッシュメモリを備えた機能ブロックである。また、図3のコンソール3 0 0自体も、太枠のブロックの部分以外は、CPU 3 0 1およびフラッシュメモリ (CMAIN) 3 0 2を備えた機能ブロックである。これらの各ブロック内のフラッシュメモリには、当該ブロックの動作を制御する制御プログラムが格納されており、当該ブロック内のCPUがその制御プログラムを実行することにより、当該ブロックが所定の機能を果たすべく動作する。必要に応じて、各ブロック間で各種の情報がやり取りされる。各ブロックの制御プログラムは、バージョン管理されており、バージョンアッププログラムによりバージョンアップすることができる。なお、「バージョン」とは、厳密には各ブロック内に設けられているフラッシュメモリに格納されている制御プログラムのバージョンを意味するが、説明の簡単化のため、ブロックのバージョンという呼び方を使う。例えば、「CTL 3 0 5のバージョン」とは、CTL 3 0 5内に設けられているフラッシュメモリに格納されている制御プログラムのバージョンのことである。

【0 0 2 2】

図3の各部について説明する。CPU 3 0 1は、CMAIN 3 0 2に格納されている制御プログラムを実行することにより、このコンソール3 0 0全体の動作を制御する。CTL 3 0 5は、CPU 3 0 1に対して各種の指示を与えてこのコンソール3 0 0の動作を制御する。本実施形態では、CTL 3 0 5のバージョンを基準として、他の各ブロックのバージョンの整合性をチェックするものとする

。LCD 3 0 6 は、CTL 3 0 5 の指示に応じて各種の情報を表示する。RAM 3 0 3 は、CPU 3 0 1 が実行するプログラムや使用する各種のデータを記憶するワーキング用メモリである。O COM 3 0 4 は、外部のコンピュータやMIDI 機器と接続するインターフェースである。O COM 3 0 4 にPC を接続し、当該PC からコンソールやエンジンやユニットの各ブロックの制御プログラムをバージョンアップすることができる。

【0 0 2 3】

PNL A 3 0 7 は、図2 で説明したセクション2 0 5, 2 0 8, 2 0 9, 2 0 7 に対応するセクション3 2 1 ~ 3 2 4 を制御するためのパネル制御用ボードである。同様にPNL B 3 0 8 は、図2 で説明したセクション2 0 1 ~ 2 0 4, 2 1 0 に対応するセクション3 2 5 ~ 3 2 9 を制御するためのパネル制御用ボードである。W COM 3 0 9 は、各種波形信号に関する通信制御を行なう部分である。例えば、モニタ用マイクやモニタ用スピーカなどの各種の入出力端子はW COM 3 0 9 に接続される。点線の矢印は波形信号の入出力を示す。COM 3 1 0 は、エンジン等との間で制御信号を授受する通信制御部である。デュアルコンソール時の2 台目のコンソールとの通信も、エンジンとの通信と同様にW COM 3 0 9 およびCOM 3 1 0 を介して行なわれる。

【0 0 2 4】

図4 に、1 台のエンジンの内部構成例を示す。エンジン4 0 0 は、CPU 4 0 1、フラッシュメモリ (EMAIN) 4 0 2、RAM 4 0 3、O COM 4 0 4、表示器4 0 5、信号処理部4 0 6、通信制御部 (B COM) 4 0 7、通信制御部 (COM) 4 0 8、および接続バス4 1 1 を備えている。信号処理部4 0 6 は、複数のDSP 4 2 1 ~ 4 2 5 を備えている。接続バ4 1 1 は、1 つのバスラインで図示したが、実際には各部が相互に情報をやり取りするための複数の方式を適用している。

【0 0 2 5】

図3 と同様に図4 でも、太枠で図示したブロックは、それぞれがその内部にCPU およびフラッシュメモリを備えた機能ブロックである。エンジン4 0 0 自体も、太枠のブロックの部分以外は、CPU 4 0 1 およびフラッシュメモリ (EM

A I N) 4 0 2 を備えた機能ブロックである。これらの各ブロック内のフラッシュメモリには、当該ブロックの動作を制御する制御プログラムが格納され、それらはバージョンアッププログラムによりバージョンアップすることができることも同じである。エンジン 4 0 0 の各ブロックのバージョン (E M A I N 4 0 2 内の制御プログラムも含む) は、C T L 3 0 5 のバージョンを基準として、バージョンチェックされる。

【 0 0 2 6 】

図 4 の各部について説明する。C P U 4 0 1 は、E M A I N 4 0 2 に格納されている制御プログラムを実行することにより、このエンジン 4 0 0 全体の動作を制御する。特に C P U 4 0 1 は、C O M 4 0 8 を介してコンソール 3 0 0 から送られる制御信号や、O C O M 4 0 4 を介して送られる制御信号に基づいて、エンジン 4 0 0 の動作を制御する。R A M 4 0 3 は、C P U 4 0 1 が実行するプログラムや使用する各種のデータを記憶するワーキング用メモリである。O C O M 4 0 4 は、外部のコンピュータや M I D I 機器と接続するインターフェースである。O C O M 4 0 4 に P C を接続し、当該 P C からコンソールやエンジンやユニットの各ブロックの制御プログラムをバージョンアップすることができる。表示器 4 0 5 は、エンジンの状態などを表示する L C D である。B C O M 4 0 7 は、コンソール 3 0 0 との間およびユニットとの間で波形信号や制御信号を授受する通信制御部である。C O M 4 0 8 は、コンソール 3 0 0 との間で制御信号を授受する通信制御部である。信号処理部 4 0 6 内の D S P 4 2 1 ~ 4 2 5 は、C P U 4 0 1 の指示に応じて、各種信号処理 (ユニットから入力した波形のミキシングや効果付与処理など) を行なう。

【 0 0 2 7 】

図 5 は、図 1 で説明したユニット 1 0 3 (1 3 3) の例である。上述したように、アナログ入力 (A I N) ユニット 5 0 1、アナログ出力 (A O U T) ユニット 5 0 2、およびデジタル入出力 (D I / O) ユニット 5 0 3 がある。これらのユニットは、それぞれ、C P U およびその C P U が実行する制御プログラムを記憶したフラッシュメモリを備えた機能ブロックである。その制御プログラムは、上述したコンソール 3 0 0 やエンジン 4 0 0 に接続した P C からバージョンア

ップできる。また、制御プログラムのバージョンは、CTL305のバージョンを基準として、バージョンチェックされる。

【0028】

図6は、この実施形態のミキサにおける信号の流れを示すブロック図である。601は、図1のユニット103、133のうち、信号入力用のユニット（図5の501、503）を示す。入力側および出力側のどちらにも複数のユニットを接続可能だが、図6では、まとめて1つの点線の矩形で示した。MADin621は、マイク信号のアナログ／デジタル変換入力ボードによる入力を示す。ADin622は、ライン信号のアナログ／デジタル変換入力ボードによる入力を示す。Din623は、デジタル入出力ボードによる入力を示す。以上の3種類のボードは、エンジンの入力側に接続されたAINユニット501またはDI/Oユニット503に差し込むことによってボード単位で増設できる。602はコンソールからの入力を示す。トークバック入力631は、コンソールのオペレータが舞台側との指示連絡用に用いる入力である。パネル入力632は、コンソールから直接入力される効果音などの波形入力である。

【0029】

入力パッチ603から出力パッチ611までの処理はエンジンが行なう。入力パッチ603は、上述した入力系統から、入力チャンネル（48ch）への任意結線を行なう。その設定は、ユーザがコンソールで所定の画面を見ながら任意に行なうことができる。入力ch604の任意の信号を、48本のMIXバス605あるいはステレオバス（Stereo_L/R）606の任意のchへ、選択的に出力できる。

【0030】

MIXバス605は、入力ch604から入力する信号をミキシングする。ミキシングされた信号は、対応するMIX出力ch609に出力される。ステレオバス606は、入力ch604から入力する信号をミキシングする。ミキシングされたステレオ信号は、ステレオ出力ch608へ出力される。ステレオ出力ch608およびMIX出力ch609の出力は、それぞれ出力パッチ611およびマトリックス出力ch610へ出力される。マトリックス出力ch610は、

ステレオ出力 c h 6 0 8 と M I X 出力 c h 6 0 9 から任意の数の信号を選択的に入力することができ、選択入力されたこれらの信号をさらにミキシングすることができる。マトリックス出力 c h 6 1 0 の出力は、出力パッチ 6 1 1 へ出力される。出力パッチ 6 1 1 は、上述した 3 種類の出力 c h 6 0 8 ～ 6 1 0 から、出力系統 (D A o u t、D o u t) への任意の結線を行なう。

【 0 0 3 1 】

6 1 6 は、図 1 のユニット 1 0 3、1 3 3 のうち、信号出力用のユニット (図 5 の 5 0 2、5 0 3) を示す。D A o u t 6 4 1 は、ディジタル／アナログ変換出力ボードへの出力を示す。D o u t 6 4 2 は、ディジタル出力ボードへの出力を示す。以上の 2 種類のボードは、エンジンの出力側に接続された A O U T ユニット 5 0 2 または D I / O ユニット 5 0 3 に差し込むことによってボード単位で増設できる。

【 0 0 3 2 】

カスケード接続部 6 0 7 は、図 1 の③で説明したようにエンジンをカスケード接続する場合のインターフェースを示す。カスケード接続部 6 0 7 を介して別のエンジンと接続することにより、バス 6 0 5、6 0 6 を 2 つのエンジンで共有できる。モニタ用セレクタ 6 1 2 とモニタ用ミキサ 6 1 3 およびコンソール 6 1 4 のモニタ用 D A o u t 6 1 5 は、各 c h の信号モニタ用の機構である。

【 0 0 3 3 】

図 7 は、コンソールの表示器 (L C D) 2 0 6 に表示されるバージョンチェックダイアログボックスの表示例を示す。バージョンチェックダイアログボックスは、ミキサの立上げ時に自動的に各部のバージョンチェックを行なった結果何らかの不整合が検出された場合、あるいはユーザがバージョンチェックスイッチを押した場合に表示される。ミキサ立上げ時にバージョンの不整合が検出されなかったときはこのダイアログボックスは表示されない。またブロックの新規接続や接続断あるいは故障などの異常が検出されたときにも、バージョンチェックを行ない、その結果がバージョンチェックダイアログボックスとして表示される。

【 0 0 3 4 】

バージョンチェックダイアログボックス 7 0 0 において、7 0 1 は、図 3 に示

したコンソール 3 0 0 の C T L 3 0 5 のバージョンおよびその他のブロック全体のバージョンを示す。「C T L 2. 4 4 O K」は、C T L 3 0 5 のバージョンが 2. 4 4 で、他のブロックと整合性が取れていることを示す。「C S 2. 4 4 O K」は、C T L 3 0 5 以外のコンソール (C S) 3 0 0 全体のバージョンが 2. 4 4 で、C T L 3 0 5 と整合性が取れていることを示す。7 0 2 は、7 0 1 で「C S 2. 4 4 O K」とまとめて表示したブロック、すなわち C T L 3 0 5 以外の各ブロックの詳細なバージョンと整合状態を示す。例えば、「C M A I N 2. 4 3 O K」は、図 3 の C M A I N 3 0 2 に格納されている制御プログラムのバージョンが 2. 4 3 で、C T L 3 0 5 と整合性が取れていることを示す。「M T R 2. 1 1 O K」は、図 3 の M T R 3 2 9 のバージョンが 2. 1 1 で、C T L 3 0 5 と整合性が取れていることを示す。以下同様にして、図 3 の太枠で示した各ブロックについて、C T L 3 0 5 とのバージョンの整合状態が表示される。なお、整合が取れていないときは、O K でなく N G と表示される。7 0 2 で表示する各ブロックのうち、1 つでも N G のブロックがあった場合は、表示 7 0 1 の C S の整合状態も N G となる。

【 0 0 3 5 】

7 0 3 は、図 4 に示したエンジン (E G) 4 0 0 の全体としてのバージョンと整合状態を示す。7 0 4 は、エンジン 4 0 0 内部の各ブロックの個別のバージョンとその整合状態を示す。7 0 4 で表示する各ブロックのうち、1 つでも N G のブロックがあった場合は、表示 7 0 3 の E G の整合状態も N G となる。7 0 5 と 7 0 6 は、エンジン 4 0 0 に接続されたユニット (図 5) のバージョンと整合状態を示す。表示 7 0 5 の中で 1 ~ 1 0 とあるのは、エンジンの入力側のユニット接続用端子を示す。ここでは 2 番目の接続用端子に A I N ユニット 5 0 1 が接続され、3 番目の接続用端子に D I / O ユニット 5 0 3 が接続されている。表示 7 0 6 の中で 1 ~ 6 とあるのは、エンジンの出力側のユニット接続用端子を示す。ここでは 3 番目の接続用端子に D I / O ユニット 5 0 3 が接続され、5 番目の接続用端子に A O U T ユニット 5 0 2 が接続されている。バージョンや整合状態の表示の仕方は、表示 7 0 1 ~ 7 0 4 と同じである。なお、- はその接続用端子にユニットが接続されていないことを示す。

【 0 0 3 6 】

7 0 7 は、図 1 の①で説明したデュアルコンソールの場合の、メインコンソール A 1 0 1 でのスレーブコンソール B 1 1 1 についての表示である。デュアルコンソールの場合のみ表示されるので点線で図示した（後述する 7 0 8 ～ 7 1 0 も同様）。表示 7 0 7 のうち「C T L 2 . 4 4 O K」はスレーブコンソール B 1 1 1 内の C T L 3 0 5 のバージョンと整合状態を示す。「C S 2 . 4 4 O K」は、スレーブコンソール B 1 1 1 内の C T L 3 0 5 以外のブロックのバージョンと整合状態を示す。これらの整合状態は、マスタコンソール A 1 0 1 内の C T L 3 0 5 のバージョンを基準とするものである。なお、スレーブ側の各ブロックの個別のバージョンと整合状態（マスタコンソールの 7 0 2 の表示と同様の表示）は、マスタ側には表示されず、スレーブコンソール B 1 1 1 の表示器に表示される。

【 0 0 3 7 】

7 0 8 , 7 0 9 は、図 1 の②で説明したエンジンのミラーリングを行なった場合の表示である。追加したエンジン B 1 2 1 について、7 0 3 , 7 0 4 と同様の表示がなされる。なお、追加したエンジン B 1 2 1 についての表示 7 0 8 , 7 0 9 における整合状態は、マスタコンソール A 1 0 1 内の C T L 3 0 5 のバージョンを基準とするものである。

【 0 0 3 8 】

7 1 0 は、図 1 の③で説明したシステムのカスケード接続を行なった場合の表示である。スレーブシステムのエンジン C 1 3 2 について、7 0 3 の表示と同様の表示 7 1 0 がなされる。カスケード接続したエンジン C 1 3 2 の整合状態は、マスタコンソール A 1 0 1 内の C T L 3 0 5 のバージョンを基準とするものである。スレーブ側のコンソール C 1 3 1 やユニット 1 3 3 については、スレーブシステムのマスタコンソール C 1 3 1 の表示器に表示される。

【 0 0 3 9 】

7 2 1 は、メッセージ表示領域である。ここではバージョンチェックの結果が正常であった場合のメッセージ表示例を図示した。7 2 2 は、動作を続行する際に押下する O K ボタンの表示である。

【 0 0 4 0 】

図 7 ではバージョンチェック結果が正常の場合の表示例を示したが、1 つでも整合しないバージョンのブロックが検出された場合にはその旨が表示される。整合しないブロックが検出された場合、それがミキサ全体の動作を継続する上で致命的となる場合と致命的とならない場合がある。図 8 の 8 0 1 と 8 0 2 は、致命的でない場合のメッセージ表示領域 7 2 1 の表示例である。8 0 1 は当該ブロックから応答が無い場合、8 0 2 は当該ブロックから応答はあったがそのバージョンが（基準のブロックのバージョンに対して）整合しない場合のメッセージ例である。8 0 1 と 8 0 2 の場合は、OK ボタン 7 2 2 が表示され、オペレータがこれを押下すると、当該ブロックとの通信を切り離して動作を継続する。強制的に動作停止するためのボタンを用意してもよい。図 8 の 8 0 3 と 8 0 4 は、致命的である場合のメッセージ表示領域 7 2 1 の表示例である。これらの場合は、OK ボタンは表示されず以後の動作を停止する。

【 0 0 4 1 】

図 9 は、本ミキサにおける状態遷移を示す。電源オン時に全ブロックが正常でバージョンの不整合もない場合は通常動作状態 9 0 1 となる。電源オン時に上記致命的なブロックの異常またはバージョン不整合が検出された場合は動作停止状態 9 0 3 となる。電源オン時に上記致命的でないブロックの異常またはバージョン不整合が検出された場合は一部動作状態 9 0 2 となる。これらの状態は、ミキサ動作中も相互に遷移する。例えば、通常動作状態 9 0 1 で正常動作している途中で、あるブロックに異常が発生したり、あるブロックが新規追加されたりしたときは、必要に応じて電源オン時と同様にバージョンチェックを行ない、その結果を表示して、一部動作状態 9 0 2 や動作停止状態 9 0 3 に遷移する。同様に、一部動作状態 9 0 2 や動作停止状態 9 0 3 のとき、異常またはバージョン不整合のブロックのボードを交換するなどして正常化することにより、通常動作状態 9 0 1 に遷移する。

【 0 0 4 2 】

以下、各ブロックのバージョンチェックについての具体例を説明する。

【 0 0 4 3 】

まず、メインシステムのコンソール300（図3）の各ブロックのバージョンチェックの具体例について説明する。メインコンソール300においては、基準となるCTL305のバージョンに対して、CMAIN302、MTR329、PNL A307、PNL B308、およびLCD306の何れかのブロック（致命的なブロック）のバージョンが不整合（または異常）である場合、動作停止状態903となる。メインコンソール300のこれら以外のブロック（致命的でないブロック）にバージョン不整合（または異常）が検出されたときは、一部動作状態902となる。

【0044】

なお、マスタコンソール300のCTL305のバージョンを基準として判断される各ブロックのバージョンの整合性の判断基準は、所定の記憶領域（例えば、CMAIN302でよい）に記憶されているものとする。例えば、図7の表示701、702では、CTLのバージョン2.44に対し、MTRのバージョンが2.11であるが、上記判断基準にCTLのバージョン2.44に整合するMTRのバージョンが記載されており、これを参照してMTRのバージョン2.11が整合していると判断している。また、致命的なブロックか否かを判断する基準についても、同様に所定の記憶領域（CMAIN302でよい）に記憶されているものとする。

【0045】

次に、ユニット（図5）のバージョンチェックの具体例について説明する。電源がオンされたとき、およびエンジンでユニットの新規接続を検出したとき、当該ユニットのバージョンチェックを行なう。エンジンのユニット接続用端子には、ユニットを接続してもしなくてもよいので、ユニットが接続されていないからといってNGにはならない。接続されたユニットのバージョンが不整合のとき、その部分をNGとしてダイアログボックス（図7）を表示し、当該ユニットとの通信を切り離して一部動作状態902となる。

【0046】

デュアルコンソールの場合のバージョンチェックの具体例について説明する。デュアルコンソールが設定されている場合、電源がオンされたとき、および第2

(スレーブ) コンソールの接続が検出されたときに、相手コンソールのバージョンチェックを行なう。相手コンソールが接続されていない場合は、NGとはせずにシングルコンソール動作（当該コンソールのみで動作続行）を行なう。相手コンソールが接続されており、かつバージョンが不整合のときは、NGとしてダイアログボックスを表示し、動作停止状態 9 0 3 となる。その後に相手コンソールが非接続になったら、ダイアログボックスを消去し、シングルコンソール動作を再開する。相手コンソールが接続されており、かつバージョンが整合するときは、OKであるので、直ちに通常動作状態 9 0 1 となりデュアルコンソール動作を開始する。デュアルコンソール動作では、まず、自分と相手のマスタスレーブ関係を確認し、問題がなければデュアルコンソール接続を確立する。マスタとマスタないしスレーブとスレーブに設定されている場合は、デュアルコンソール接続を行なわない。デュアルコンソール接続が確立された場合、第 1 コンソールと第 2 コンソールの両方で、接続されたエンジンのミキシング処理を制御することができる。

【 0 0 4 7 】

ミラーリングの場合のバージョンチェックの具体例について説明する。ミラーリングが設定されている場合、電源がオンされたとき、および第 2 エンジンの接続検出時に、第 2 エンジンのバージョンチェックを行なう。バージョンチェック動作は、基本的に上記デュアルコンソールの場合と同じであり、整合しないエンジンが接続されている間は動作停止状態 9 0 3 になる。ミラー動作では、2 つのエンジンにマスタスレーブ関係は無いので、バージョンさえ整合すれば直ちに通常動作状態 9 0 1 となりミラー接続が確立される。ミラー接続が確立された場合、2 つのエンジンは、コンソールからの制御により、相互に同じ動作を行なう。選択されているエンジンが故障したとき、自動ないし手動で他方のエンジンに切り換えられる。

【 0 0 4 8 】

カスケードの場合のバージョンチェックの具体例について説明する。カスケードが設定されている場合、電源がオンされたとき、および第 2 エンジンの接続検出時に、第 2 エンジンのバージョンチェックを行なう。バージョンチェック動作

は、基本的に上記デュアルコンソールの場合と同じである。カスケード動作では、まず、自分と相手のマスタスレーブ関係を確認し、問題がなければ通常動作状態 9 0 1 となりカスケード接続を確立する。マスタとマスタないしスレーブとスレーブに設定されている場合は、カスケード接続を行なわない。カスケード接続が確立された場合、2つのミキサシステムのミキシングバスが連結され、全体として c h 数が倍のミキサシステムとなる。

【 0 0 4 9 】

なお、上記ではバージョンチェックの例を説明したが、各ブロックで異常が検出された場合（例えば、何れかのブロックが接続断の状態になったり故障が検出された場合など）、そのブロックは N G になるので、バージョンチェックダイアログボックスの表示データは書き換えられる。例えば、I N C で異常が検出されたら（これは致命的でない異常である）、コンソール 3 0 0 のメイン C P U 3 0 1 は、I N C 3 2 7 に対して動作停止を指示するとともに、P N L _ B 3 0 8 に対して I N C 3 2 7 との通信停止を指示する。また、図 7 の表示データ中、I N C の状態を O K から N G に書き換える。別の例として、例えば、C T L 3 0 5 で異常が検出されたら（これは致命的な異常である）、コンソール 3 0 0 のメイン C P U 3 0 1 は（接続されているコンソール、エンジン、ユニットを含めて）全ブロックに対して動作停止を指示する。

【 0 0 5 0 】

図 1 0 は、図 3 に示したコンソール 3 0 0 のメイン C P U 3 0 1 のパワーオン時の処理を示す。ステップ 1 0 0 1 で、初期設定をし、ステップ 1 0 0 2 で自コンソールと接続されている第 1 エンジンの各ブロックのバージョンのチェックを行なう。なお、任意のブロックのバージョンは、そのブロックに問合せを発行することで得ることができる。ステップ 1 0 0 3 でデュアルコンソールになっているか否か判定し、デュアルコンソールの場合は、ステップ 1 0 0 4 で第 2 （スレーブ）コンソールの各ブロックのバージョンのチェックを行なう。ステップ 1 0 0 5 でミラーがオンされているか否か判定し、ミラーオンの場合は、ステップ 1 0 0 6 で第 2 エンジンの各ブロックのバージョンをチェックする。次に、ステップ 1 0 0 7 でカスケードがオンされているか否か判定し、カスケードオンの場合

は、ステップ1008で第2（スレーブ）システムの各ブロックのバージョンチェックを行なう。ステップ1009では、以上の各バージョンチェックで不整合（または異常）があるか否か判定する。不整合（または異常）が無いときは通常動作状態に移行し動作を継続する。不整合（または異常）があったときは、ステップ1010で致命的であるか否か判定し、致命的な場合は、ステップ1011で、接続されている各ブロックの動作停止を指示し、ステップ1012で、図7で説明したダイアログボックスを表示し、動作停止状態に移行する。致命的でないときは、ステップ1013で、問題があるブロックの停止と通信の切り離しを指示し、ステップ1014で、ダイアログボックスを表示し、一部動作状態に移行する。

【0051】

図11は、図10のステップ1002，1004，1006，1008（および後述のステップ1201，1211）で実行するバージョンチェックの処理の詳細を示す。ステップ1101で、チェック対象のブロックのバージョン情報を収集する。ステップ1102で収集が完了したか否か判定し、完了していたらステップ1105に進む。バージョン情報の収集が完了していないときは、ステップ1103でタイムアウトするまでステップ1101からの処理を繰り返す。タイムアウトのときは、ステップ1104で、収集できなかった不足分の情報は取得不能と判定し、ステップ1105に進む。ステップ1105では、収集したバージョン情報が、基準となるCTLのバージョンに整合しているか否か判定する。

【0052】

図12（a）は、コンソールがエンジンないし2台目のコンソールの新規接続を検出したときの処理を示す。ステップ1201で、新規に接続された機器の各ブロックのバージョンをチェックする。ステップ1202で、それらのバージョンが整合しているか否か判定する。整合しているときは通常動作状態に移行する。整合していないときは、ステップ1203で、接続されている各ブロックの動作停止を指示し、ステップ1204で、バージョンチェックダイアログボックスを表示し、新規に接続されたエンジンないしコンソールに関する表示を行なう。

すなわち、エンジンが接続された場合は、それが1台目であればバージョンの整合状態を表示703および704に表示する。2台目のエンジンの接続はミラーオンに設定されている場合のみ検出され、その整合状態が表示708および709に表示される。また、2台目のコンソールの接続はデュアルコンソールオンの場合のみ検出され、その整合状態が表示707に表示される。ステップ1204の後、動作停止状態に移行する。すなわち、本実施形態では、コンソールとコンソール、ないしコンソールとエンジンで、バージョンの不整合が発生した場合は、それを致命的であると判断し、動作停止状態903に入るのである。この場合でも、不整合の発生したコンソールないしエンジンを切り離すことにより、通常動作状態901に復帰できる。

【0053】

図12(b)は、ユニットの新規接続を検出したときの処理を示す。ステップ1211で、接続されたユニットのバージョンチェックを行なう。バージョンが整合していたときは、ステップ1212から通常動作状態に移行する。整合していないときは、ステップ1213で、該ユニットとの通信の切り離しを指示し、ステップ1214で、バージョンチェックダイアログボックスを表示し、新規に接続されたユニットに関する表示を行なう。すなわち、入力側のユニットが接続された場合には、表示705のそのユニットを接続した端子位置にバージョンの整合状態が表示され、出力側のユニットが接続された場合には、表示706の接続端子位置に表示される。ステップ1214の後、一部動作状態に移行する。すなわち、エンジンとユニットとの間のバージョンの不整合が発生した場合、それは致命的でないと判断され、不整合の発生したユニットが切り離されて一部動作状態に入る。

【0054】

図12(c)は、各機器との接続断を検出したときの処理を示す。ステップ1221では、画面表示等により、接続が切れたことをユーザに通知する。ステップ1222では、接続の切れた機器が何であるかに応じてシステムの状態を更新する。例えば、2台のコンソールが接続されデュアルコンソールの動作をしている状態でコンソールBとの接続が切れた場合、コンソールAのみのシングルコン

ソールの状態に移行する。また、バージョンが不整合のユニットが物理的にエンジンから切り離された場合には、そのユニットに関する不整合が解消される。さらにまた、通常動作状態にあるコンソールとエンジンの間が切断された場合には、コンソールとしての通常動作は可能だがシステムとしてのミキシング動作は不能な状態に移行する。

【 0 0 5 5 】

次に、ステップ 1 2 2 3 では、接続断が検出された時にバージョンチェックダイアログボックスを表示中であった場合、ないし、ステップ 1 2 2 2 の更新後も何らかの不整合が残っている場合にはステップ 1 2 2 4 に進み、それ以外の場合には通常動作状態へ移行する。

【 0 0 5 6 】

ステップ 1 2 2 4 では、バージョンチェックダイアログボックスを表示し、接続の切れた機器に関する表示部分を更新する。例えば、デュアルコンソールのコンソール B との接続が切れた場合には、相手側のコンソールに関する表示 7 0 7 を消去する。また、エンジンからユニットが切り離された場合は、表示 7 0 5 ないし 7 0 6 におけるそのユニットの表示部分を - に変更する。また、コンソールとエンジンが切り離された場合は、接続断となったエンジンの表示 7 0 3 を NG とする。ステップ 1 2 2 5 では、バージョンチェックの結果に応じて、不整合が全くなければ通常動作状態に、一部不整合でも動作可能であれば一部動作状態に、不整合が致命的であれば動作停止状態に移行する。

【 0 0 5 7 】

このように、図 7 のダイアログボックスは、バージョンチェックが行なわれ、その結果バージョンの不整合が検出されたときに自動的に開かれる。また、ユーザがマウスによりバージョンチェックダイアログを呼び出す操作を行なうことも可能である。

【 0 0 5 8 】

なお、本実施形態は、コンソール、エンジン、およびユニットが分離されたミキサで説明したが、全てが一体のミキサや、コンソールとエンジン分離型のミキサに適用しても良い。また、バージョンチェックの基準は C T L にしたが、どの

ブロックを基準にしてもよい。

【 0 0 5 9 】

本実施形態では、制御プログラムのバージョンをその機能ブロックのバージョンとして説明したが、機能ブロックのボード自体にバージョンが付与されている場合でも本発明は適用可能である。例えば、機能ブロックとなるボードがCPU無しのハードウェアで構成されている場合、CPUを含むハードウェアとファームウェアの組み合わせで構成されている場合などは、ボード自体にバージョンが付けられているので、そのバージョンの整合性をチェックするようにしてもよい。

【 0 0 6 0 】

本実施形態のデュアルコンソール動作では、スレーブコンソールのバージョンが不整合であった場合、ミキサシステム（マスタコンソール＋エンジン）が動作停止状態903に入るようになっていたが、シングルコンソールとして動作を継続する（一部動作状態902）ようにしてもよい。同様に、ミラーリング動作でも、接続された2台のエンジンのうち的一方が不整合であった場合に、動作停止状態903に入らずに、バージョンが整合する方のエンジンとコンソールで動作を継続する（一部動作状態902）ようにしてもよい。

【 0 0 6 1 】

【発明の効果】

以上説明したように、この発明によれば、電源立ち上げ時に、システムの全ブロックのバージョンの整合性をチェックし、不整合がなければ動作を継続し、不整合があった場合はユーザに警告するようにしているので、バージョンの不整合を電源立ち上げ時に即時に確認できる。また、不整合が致命的である場合は動作を停止し、致命的でない場合はその不整合を無視して動作を継続する旨の通知をした後、不整合と判定されたブロックを切り離して動作を継続するので、最低限動作を継続できる場合は継続することができる。新規に機器を接続した場合や故障などの異常が発生した場合も、致命的な不整合が起きていなければ、一部動作状態で継続動作するので、制限があっても動作継続したい場合には便宜である。

【図面の簡単な説明】

【図 1】 この発明の実施の形態に係るデジタル・ミキサの構成例を示す
ブロック図

【図 2】 1 台のコンソールの外部パネルの例を示す図

【図 3】 1 台のコンソールの内部構成例を示すブロック図

【図 4】 1 台のエンジンの内部構成例を示すブロック図

【図 5】 ユニットの例を示す図

【図 6】 ミキサにおける信号の流れを示すブロック図

【図 7】 バージョンチェックダイアログボックスの表示例を示す図

【図 8】 メッセージ表示領域の表示例を示す図

【図 9】 本ミキサにおける状態遷移を示す図

【図 1 0】 パワーオン時の処理を示すフローチャート図

【図 1 1】 バージョンチェックの処理のフローチャート図

【図 1 2】 新規接続や接続断を検出したときの処理のフローチャート図

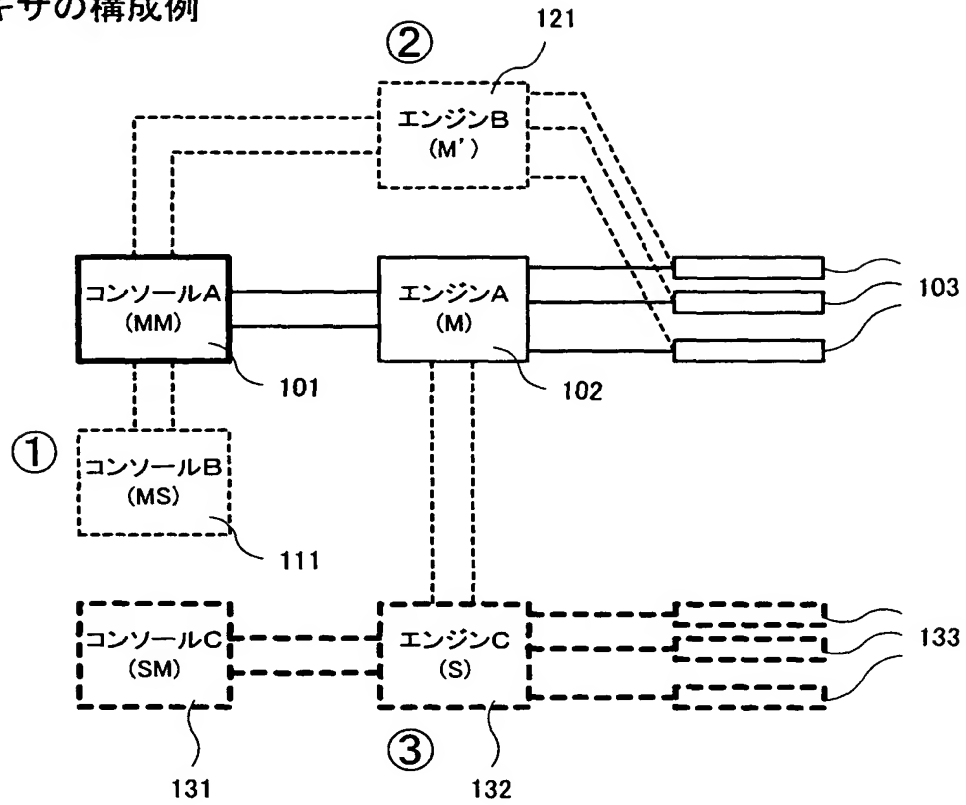
【符号の説明】

1 0 1 …コンソール、 1 0 2 …エンジン、 1 0 3 …各ユニット。

【書類名】 図面

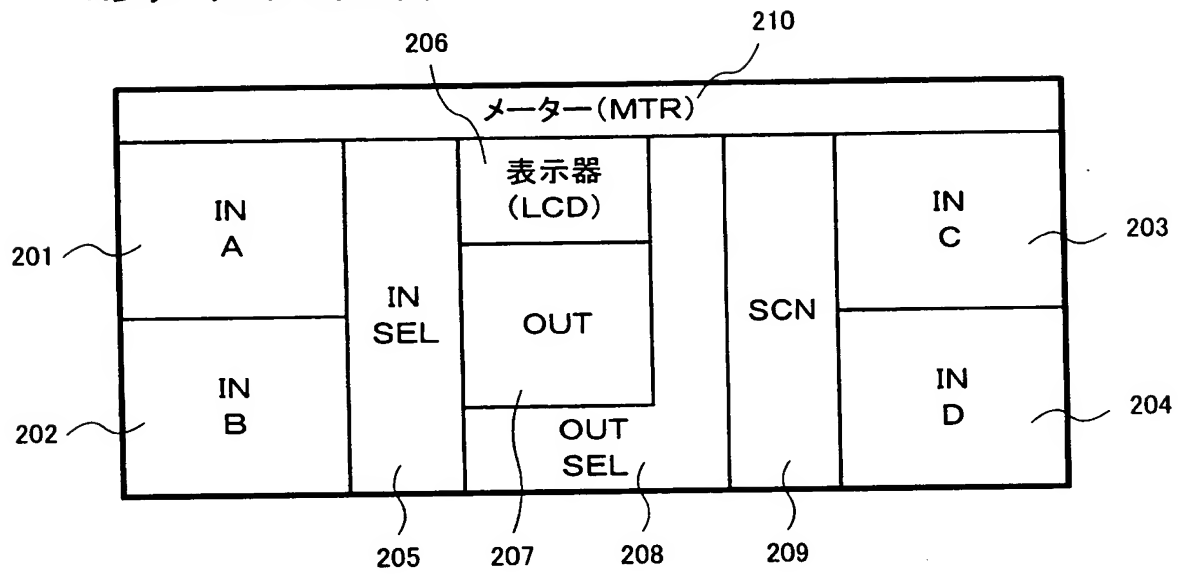
【図 1】

ミキサの構成例

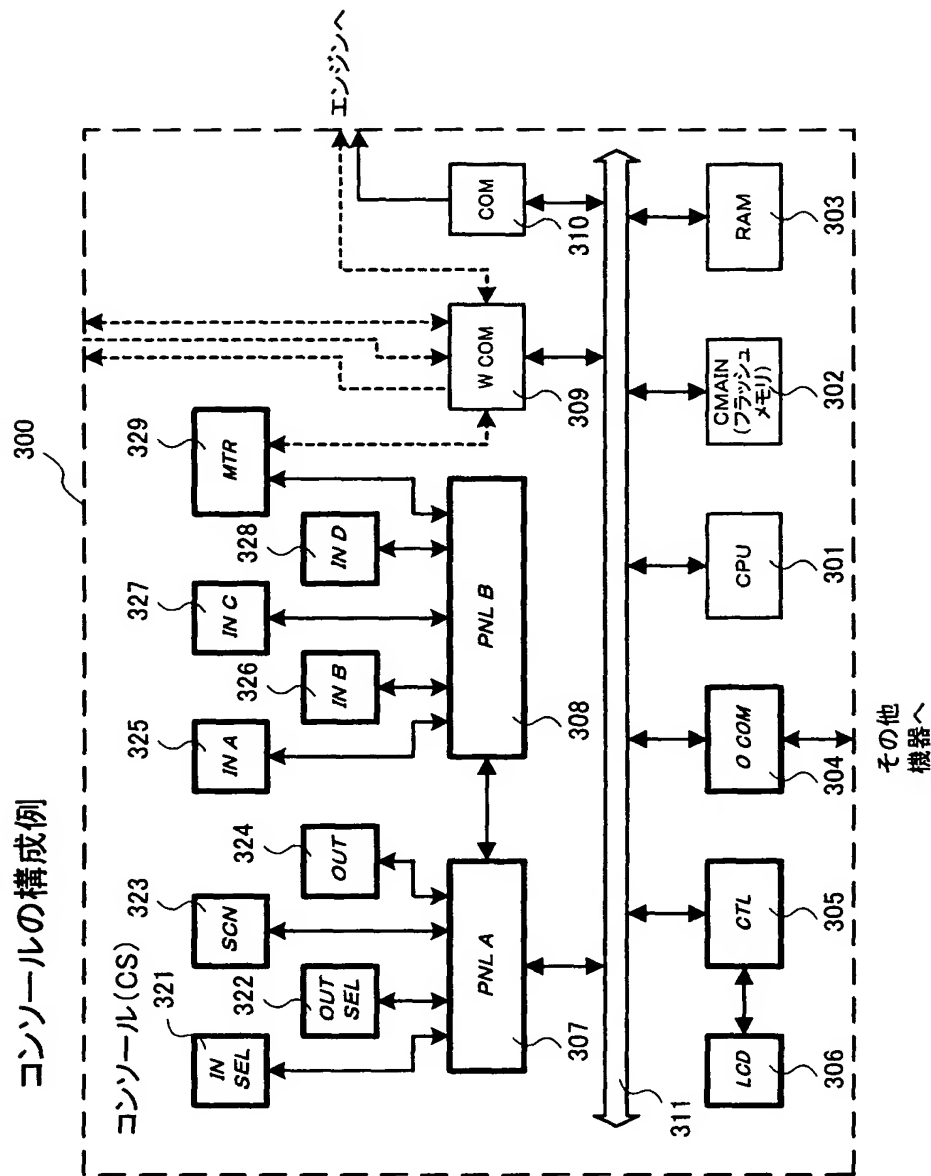


【図 2】

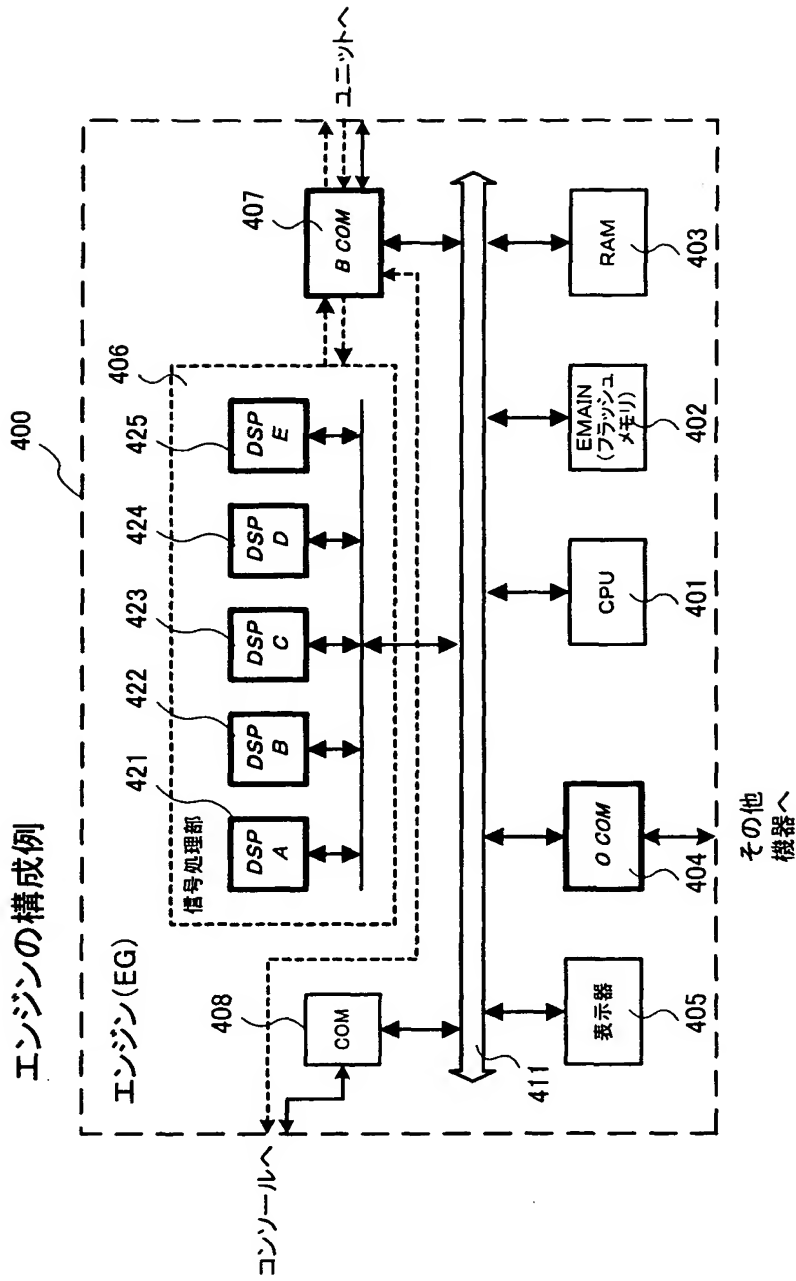
コンソールのパネル図



【図 3】

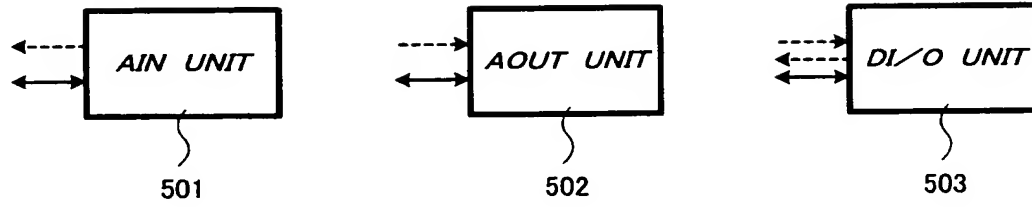


【図 4】

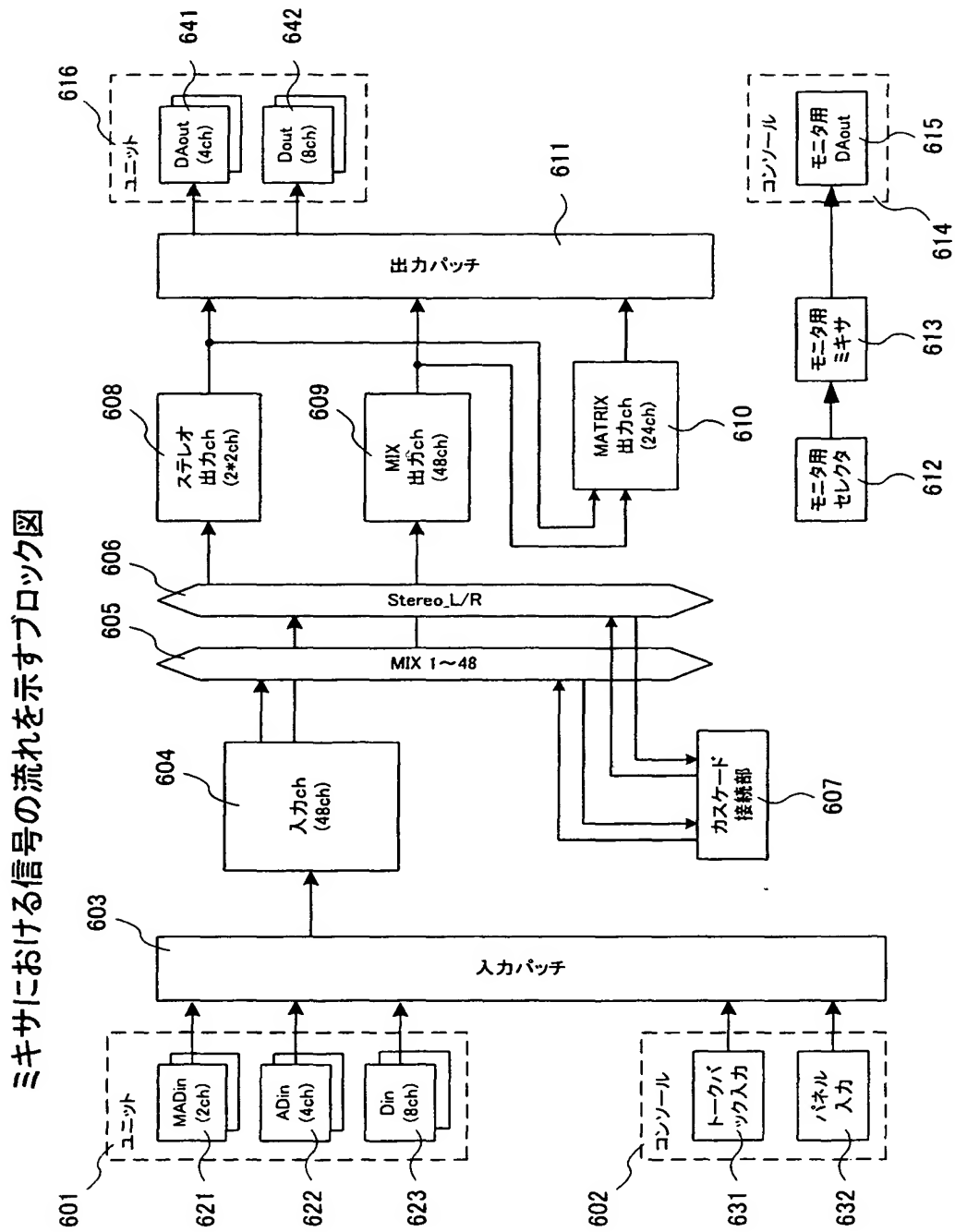


【図 5】

ユニットの構成例

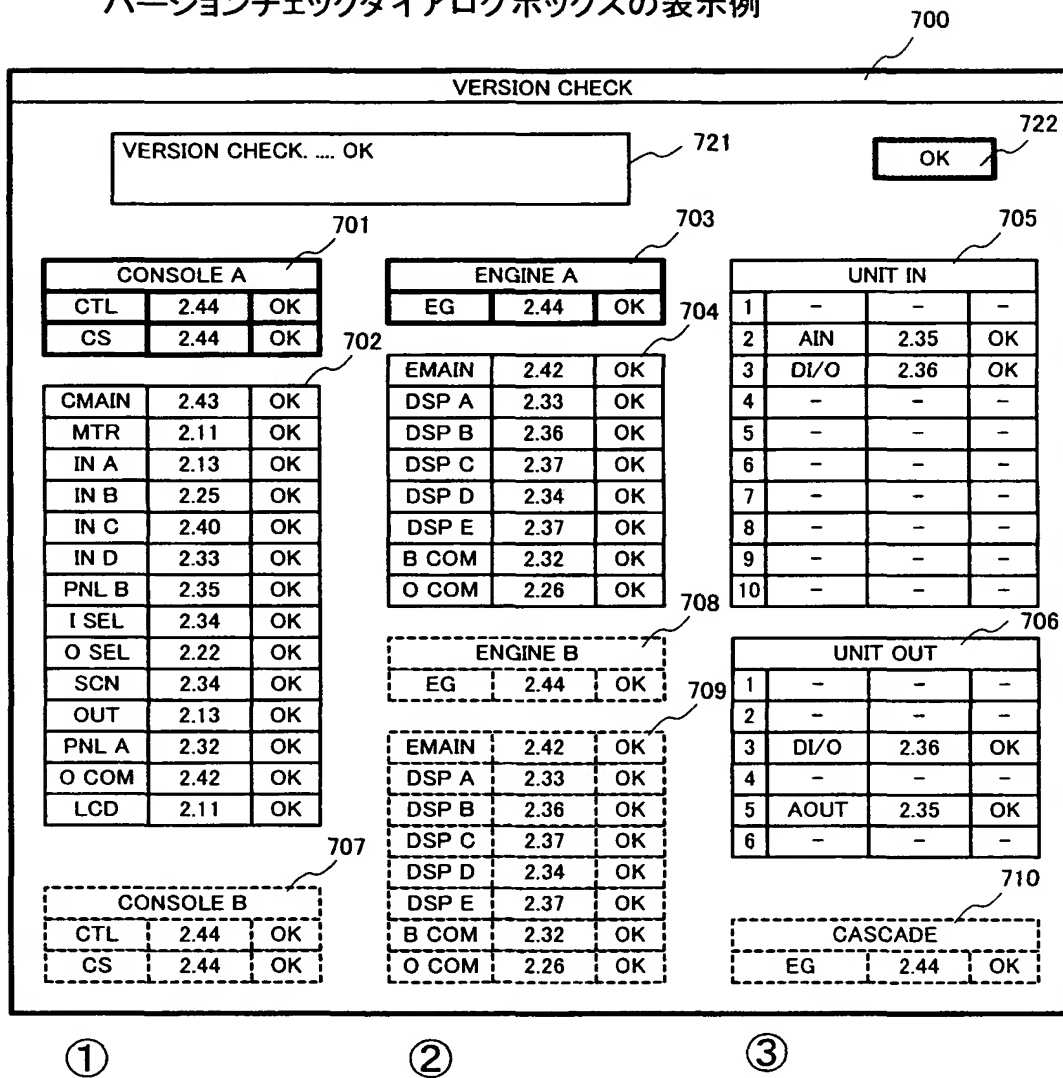


【図 6】

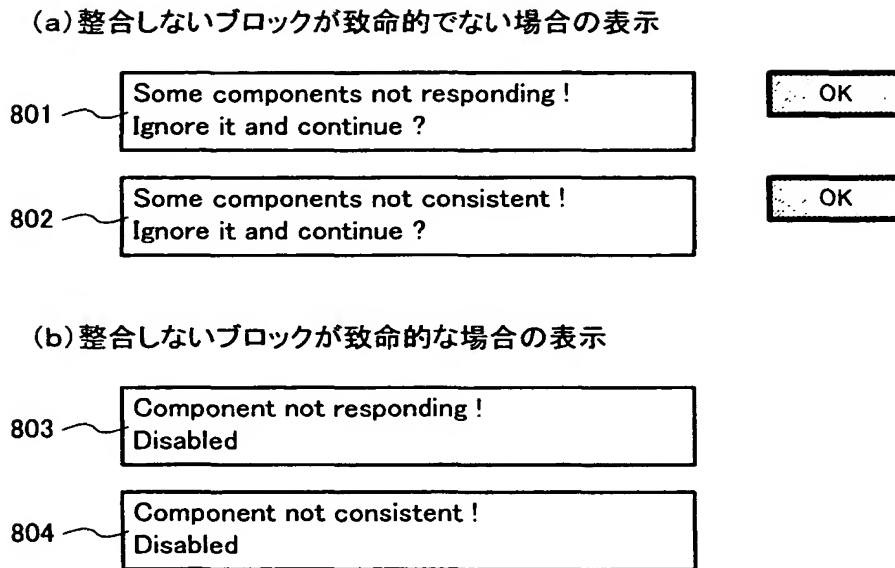


【図 7】

バージョンチェックダイアログボックスの表示例

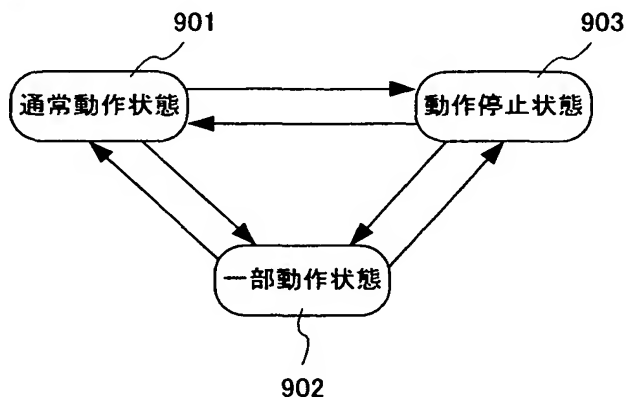


【図 8】



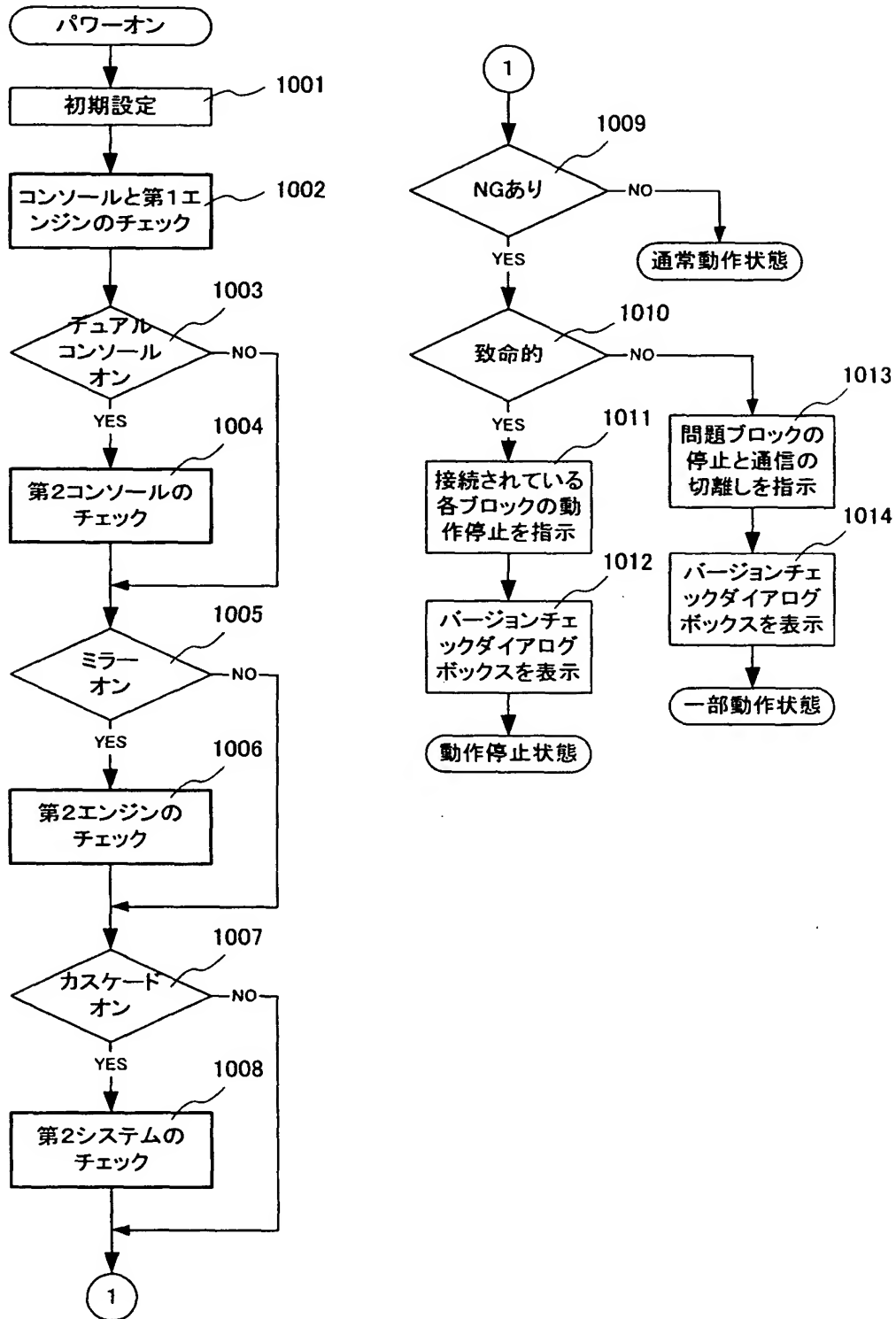
【図 9】

状態遷移図



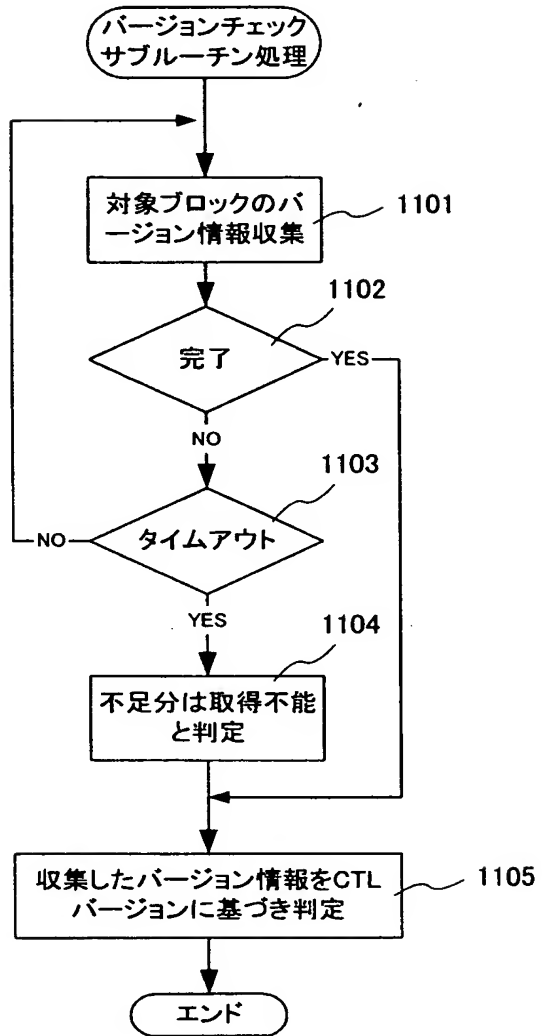
【図 1 0】

コンソールのCPUの処理

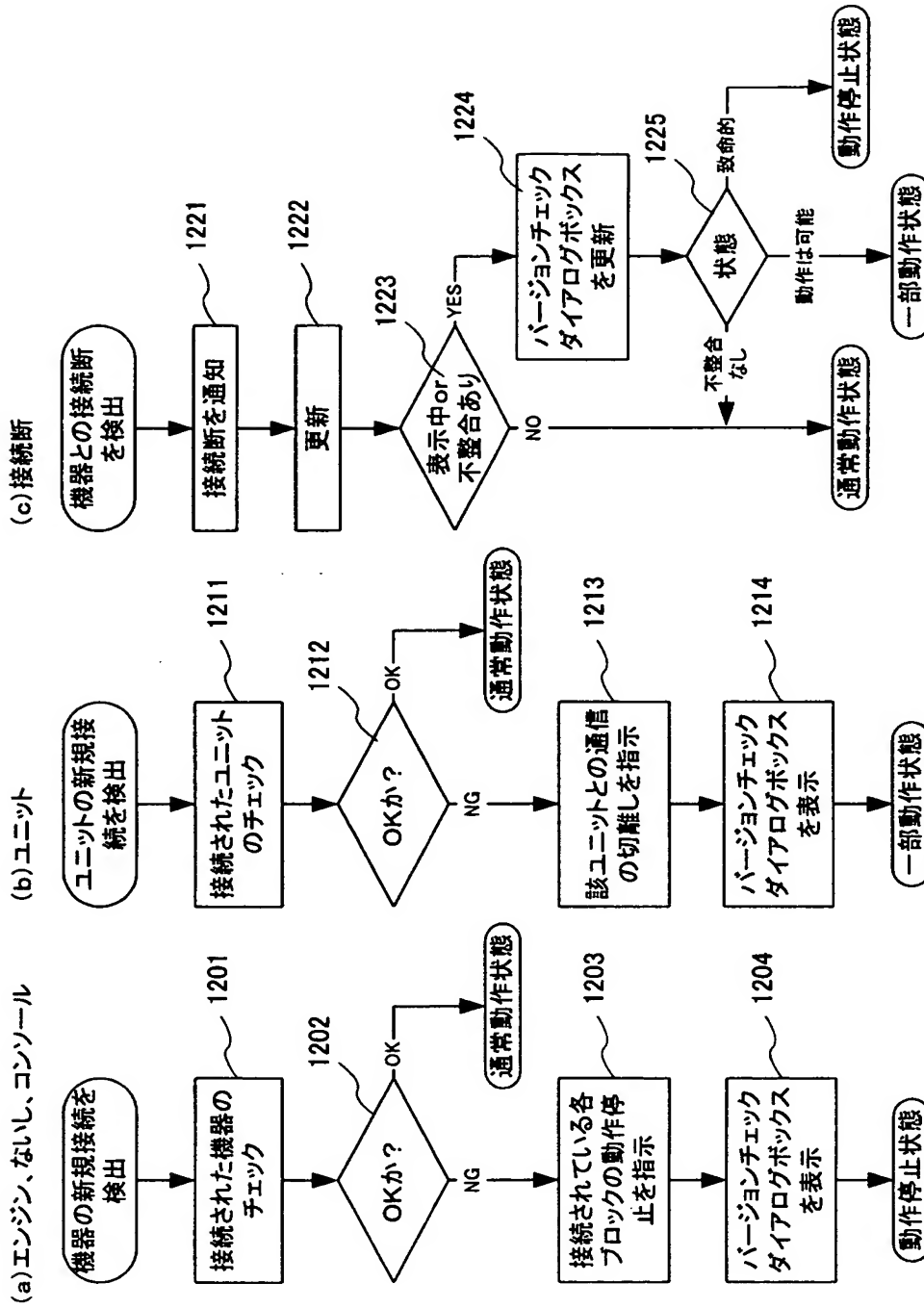


【図 1 1】

バージョンチェック処理



【図 1 2】



【書類名】 要約書

【要約】

【課題】

複数の機能ブロックを備え、各機能ブロック毎にCPUとそのCPUが実行する制御プログラムを備えているようなデジタル・ミキサにおいて、それら各機能ブロック毎のCPUで動作する制御プログラムのバージョンを、適正にかつ容易に管理できるようにすることを目的とする。

【解決手段】

電源立ち上げ時に、システムの全ブロックのバージョンの整合性をチェックし、不整合がなければ動作を継続し、不整合があった場合はユーザに警告する。その警告において、不整合がシステムの動作にとって致命的である場合は動作を停止し、致命的でない場合はユーザに「バージョンの不整合を無視して動作を継続する」という内容の通知をした後、不整合と判定されたブロックを切り離して動作を継続する。

【選択図】 図 1 0

認定・付加情報

特許出願の番号	特願 2 0 0 2 - 2 1 0 3 1 0	
受付番号	5 0 2 0 1 0 5 8 2 5 0	
書類名	特許願	
担当官	第八担当上席	0 0 9 7
作成日	平成 1 4 年	7 月 1 9 日

<認定情報・付加情報>

【提出日】 平成14年 7月18日

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 4 0 7 5]

1. 変更年月日	1 9 9 0 年 8 月 2 2 日
[変更理由]	新規登録
住 所	静岡県浜松市中沢町 1 0 番 1 号
氏 名	ヤマハ株式会社